Significance of Object Size Based Caching Algorithms for Personalized Web Applications

S. R. Balasundaram
Assistant Professor,
Department of Computer Applications,
National Institute of Technology, Tiruchirappalli-620 015. India.
blsundar@nitt.edu

S. Akhilan
Research Scholar,
Department of Computer Science and Engineering,
National Institute of Technology, Tiruchirappalli-620 015. India.
akhilan@nitt.edu

Abstract

Web application is an application that is accessed through a web browser over a network. The network can be internet or intranet. Personalizing the content of web pages for different web applications is a challenging task in present web scenarios. Web caching play a vital role in delivering the web contents to the users in a resourceful way. Hence, web caching is a technique which is universally accepted for reducing the consumption of internet bandwidth caused due to drastic World Wide Web (WWW) growth. Personalization and speed of the web content delivery is an important factor to be considered. This depends on the object size and the network traffic that exists in the WWW. In this paper we study the performance of page replacement strategies called Class-Based LRU (C-LRU) and Pass Down C-LRU (PD-C-LRU).

Keywords: C-LRU, PD-C-LRU, Personalization, Web caching.

1. INTRODUCTION

As the internet plays a vital role in the present scenario, the users intended to utilize the resources available in the web increases rapidly. This causes network traffic which slows down the activities of the web browsers. The presence of network traffic in the internet originates from WWW requests. The increasing use of WWW based services has not only led to higher technological web servers but also heavily used components in the Internet. So, it is very appropriate to use some techniques to tackle the network traffic. Some web sites which are popular are frequently used by web browsers. It is very appropriate to use caching techniques at this situation and to overcome the network traffic which exists in the internet. When a request is satisfied by a cache, the content no longer has to travel across the internet from the origin web server. This will save the bandwidth for the cache user as well as for the main server.

Personalization plays the role of tailoring the web contents as required by the users. While performing the process of
content delivery as per the user requirements, caching is one of the factors considered. The heart of a caching system is its page replacement policy. Web caching is one of the renowned techniques to decrease the end-to-end delays that exist in the WWW applications. Web caching performs similar functions like memory system caching, where the cache stores web pages by anticipating the future requests. However, there exists a significant difference between memory system and web caching in terms of object sizes, retrieval costs, and cache ability [2]. Web caching is used by the user to increase the bandwidth availability by limiting the transmission of redundant data, to reduce the network congestion, to improve the response times and to achieve lower cost of bandwidth.

Web caching has some special properties that make it an interesting and pay way for new methodologies. It is different from that of traditional approaches towards caching such as the sizes of the objects to be cached vary greatly, the costs to request particular objects differ largely and objects in the cache are read only and hence no write back mechanism is required. There are three possible locations of the cache namely at the client side, at the proxy server side and at the primary web server side in the present web set-up [1, 3].

2. EXISTING CACHING STRATEGIES

Over the last few years, many well-known caching strategies have been developed (Figure 1). The aim of these different strategies is to improve the cache hit rate which is defined as the percentage of requests that could be served from the cache and the cache byte hit rate defined as the percentage of bytes that could be served. All these algorithms were developed for specific contexts and it is shown that an algorithm which is optimal for one context may fail to provide good results for another context.

Williams et al. and Abrams et al. (2000) have compared LRU and FIFO with LFF () strategy [7]. P. Jelenkovi et al. (2008) have proposed a class of randomized LRU caching algorithms which are used to alleviate the effect of variable document sizes on Web caching [4]. Hua Chen et al. (2006) have proposed a new prediction algorithm for deciding upon which page should be pre-fetched [5]. Piyush Kumar et al. (2004) have proposed a model that assumes a cache line chosen for replacement is the one that will be accessed in the future [6].

Fig 1. Cache diagram

In this paper the significance of two LRU algorithms namely Class Based LRU (C-LRU) and Pass Down Class Based LRU (PD-C-LRU) is discussed.

3. CLASS BASED LRU (C-LRU)

The caching strategy class-based LRU (C-LRU) is the extension/enhancement of standard LRU replacement policy. In all page replacement strategies, object size
distributions in the WWW are tightly coupled i.e. even though small objects are more popularly used by most of the web users and are requested more number of times for various web applications, large objects can also occur more often than it has been expected for the web applications.

The C-LRU caching algorithm depends on size based and as well as frequency based [8, 9, 10]. In most of the caching methods, the object sizes are ignored entirely or either small or large objects are considered for web caching. Since, caching of large objects increases the byte hit rate and decreases the hit rate, both a high byte hit rate and a high hit rate can only be attained by creating a proper balance between large and small objects that have to be stored in the cache.

When C-LRU replacement technique is used, the cache is divided into different portions kept for objects of a different size. Each portion is called as class which is reserved for storing documents of specific and different size (Figure 2). The memory capacity provided for the cache is divided into ‘W’ partitions. Among this ‘W’ partitions, each partition (where \( w = 1 \ldots W \)) takes a specific fraction ‘\( w_x \)’ of the cache (\( 0 < w_x <1 \), \( M, \sum x w_x = 1 \)), where summation can be expanded from \( x=1 \) to \( W \). Partition ‘\( x \)’ caches objects belonging to class \( x \), where class \( x \) is defined to include all objects of size ‘\( S_i \)’ with \( h_{x-1} \leq S_i < h_x (0 = h_0 < h_1 < \ldots < h_{X-1} < h_x = \infty) \). Thus, when an object has to be cached, its class (Figure 1) has to be determined before it is passed to the corresponding partition. For determining the class, it is essential to determine the values of \( m_1, \ldots m_X \) and \( w_1, \ldots w_X \).

Since, the reserved cache for large documents was not large enough to encompass the assigned files, it is not possible to fit the object of larger size into the cache memory of C-LRU replacement strategy. Solution to this problem is achieved by using PD-C-LRU replacement policy [11].

4. **PASS DOWN – CLASS BASED LRU (PD-C-LRU)**

Fitting larger size objects into cache is one of the essential requirements in web environments including personalized applications. In PD-C-LRU replacement strategy, knowing the overall cache size the memory size of individual classes (i.e. class 1 to 5) can be known. For example, if the entire cache memory size is 417 KB then the reserved cache for class 5 is 20.85%. Since the fifth class is used to store documents with large size, many of the documents will not fit at this class. This implies that the fifth class might be kept null most of the time (Figure 3).

Assume that a new document to be stored in the cache, starting from class 1. But class 1 is completely filled, so that one document has to be removed from it, although there is enough storage places in
rest of the classes (especially in class 4 and 5). The main idea of the PD-C-LRU caching strategy is shown in figure 4, which predicts the function/operation of PD-C-LRU caching strategy. Each partition is divided into ‘fit’ portion (indicated as red color box) and ‘unfit’ portion (indicated as white color box). The ‘fit’ portion is used by C-LRU and caches the documents with appropriate sizes and the ‘unfit’ portion area of a class might include documents from all others i.e. files with large size. Instead of removing the least recently used document from a class, we try to place it in the ‘unfit’ segment of the next class i.e. pass-down approach. If the cache is completely filled and a new document has to be cached then the documents from the ‘unfit’ regions are removed from class 5 onwards so that new files with large size files are made fit into the cache access place.

5. EXPERIMENT AND RESULTS

A. Screen Shots

Figure 5 and 6 illustrate the experimental results of the program written using C++ programming.
Significance of Object Size Based Caching Algorithms for Personalized Web Applications

B. Graph Comparison

The performance of C-LRU and PD-C-LRU is compared based on the object size. From figure 7, it is understood that PD-C-LRU is better than C-LRU page replacement strategy.

6. CONCLUSION

Web applications play an important role in present web world. Though different caching methods are available, web caching
is predominant among web users. Personalization and web caching deliver the required content to the users from that of the vast available informations present in the internet. The page replacement algorithms namely C-LRU and PD-C-LRU are compared and the significance of PD-C-LRU is highlighted.

REFERENCES


